AMENDMENTS TO THE SPECIFICATION:

Please replace the paragraph beginning at line 1 on page 1 and ending at line 3 on page 1 with the following amended paragraph:

A1

This application claims priority to a provisional patent application No. 60/163,851~~TBD~~, entitled "A Method for Iterative Joint Optimization of Language Model Perplexity and Size", filed on 11/5/99 by the inventors of this application.

Please replace the paragraph beginning at line 23 on page 3 and ending at line 26 on page 3 with the following amended paragraph:

A2

As a result of the foregoing limitations, a language model using prior art lexicon and segmentation algorithms tends to be error prone. That is, any errors made in the lexicon or segmentation stage are propagated throughout the language model, thereby limiting its accuracy and predictive attributes.

Please replace the paragraph at lines 8-12 on page 15 with the following amended paragraph:

A3

Frequency calculation function 213 identifies a frequency of occurrence for each item (character, letter, number, word, etc.) in the training set subset. Based on inter-node dependencies, data structure generator 21~~8~~0 assigns each item to an appropriate node of the DOMM tree, with an indication of the frequency value (C_i) and a compare bit (b_i).

Please replace the paragraph at lines 1-18 on page 22 with the following amended paragraph:

Once the training data is combined, LMA 104 performs language model compression based, at least in part, on memory and/or application constraints. More particularly, once controller 202 has developed an initial language model by combining the training data, the language model is iteratively refined to accommodate the size constraints while minimizing any adverse affect on language model performance, block 510. According to one aspect of the present invention, controller 202 utilizes a relative entropy-based pruning algorithm. Conceptually, controller 202 removes as many un(der)-utilized probabilities as possible without increasing the language model perplexity. In this regard, controller 202 examines the weighted relative entropy between each probability P(w|h) and its value P'(w|h') from the backoff distribution. Mathematically, the this distance is expressed as follows:

$$D(P(w|h), P'(w|h')) = P(w|h) * \log(P(w|h)/P'(w|h')) \qquad (5)$$

As used herein, h is a history and h' is a reduced history. For small distances the backoff probability itself is good approximation and P(w|h) does not carry much additional information. In such a case, controller 202 deletes this probability from the model. The deleted probability mass is reassigned to the backoff distribution, and controller 202 recalculates the backoff weights.

A4

Please replace the paragraph at lines 1-15 on page 26 with the following amended paragraph:

The combining counts method begins at block 902, whereinien controller 202 determines the count (frequency) for each of the disparate training chunks. According to one implementation, controller 202 calculates the count using the equation:

#5

$$C(w_i, w_{i-1}, w_{i-2}) = \lambda_j C^j(w_i, w_{i-1}, w_{i-2}) \quad (7)$$

In the equation above, $C^j(w_i, w_{i-1}, w_{i-2})$ is the count of the trigram within training chunk j, and is weighted with a measure of the training chunk perplexity (λ_j). The weighting measure λ_j is defined by the following equation:

$$\lambda_j = (1/PP_j)/(1/PP_o) \quad (8)$$

That is, the weighting value is the ration of the perplexity of the training chunk j to the perplexity of the tuning set.

Please replace the paragraph at lines 1-4 on page 27 with the following amended paragraph:

A6

Once clustered, controller trains a language model for each of the clusters. According to one implementation, controller 202 invokes an instance of Markov probability calculator 212 to generate the cluster language models. In alternate embodiments, N-gram language models are utilized by controller 202.

Please replace the paragraph at lines 9-17 on page 27 with the following amended paragraph:

A7

According to one embodiment, the resultant cluster language models are merged to form a composite language model "mixture". Initial experimental results indicate that such a merging may be performed without a degradation in LM performance, and offers computational decoding advantages. According to one embodiment, the language model is merged using linear interpolation. That is, the probability of a word, w, is the linear interpolation of 2 merge LMs. ~~I~~i.e., P(w)=P1(w)+alpha*P2(w), where P1(w) is the probability from LM1, P2(w) is from LM2, and where alpha is the interpolation weight, which is estimated by using EM algorithm.